Instance scale, numerical properties and design of metaheuristics

A study for the facility location problem

Chalupa, David; Nielsen, Peter

Published in:
IFAC-PapersOnLine

DOI (link to publication from Publisher):
10.1016/j.ifacol.2019.11.535

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 52-13 (2019) 2219–2224

Instance Scale, Numerical Properties and Design of Metaheuristics: A Study for the Facility Location Problem

David Chalupa and **Peter Nielsen**

Operations Research Group
Department of Materials and Production
Aalborg University
Fibigerstræde 16, Aalborg 9220, Denmark
{*dc,peter*}*@m-tech.aau.dk*

Abstract: We present an in-depth computational study of two local search metaheuristics for the classical uncapacitated facility location problem. We investigate four problem instance models, studied for the same problem size, for which the two metaheuristics exhibit intriguing and contrasting behaviours. The metaheuristics explored include a local search (LS) algorithm that chooses the best moves in the current neighbourhood, while a randomised local search (RLS) algorithm chooses the first move that does not lead to a worsening. The experimental results indicate that the right choice between these two algorithms depends heavily on the distribution of coefficients within the problem instance. This is also put further into context by finding optimal or near-optimal solutions using a mixed-integer linear programming problem solver.

Keywords: facility location problem, local search, combinatorial optimisation, integer linear programming, algorithm efficiency

1. INTRODUCTION

Many problems in science and engineering are widely regarded as computationally hard. Within operations research, these involve a number of planning, scheduling or production optimisation problems. Such problems include a variety of facility location (Gao and Robinson, 1992), supply chain optimisation problems (Melo et al., 2009);(Sitek et al., 2014), as well as shop scheduling problems (Blum and Sampels, 2004), including job shop scheduling (Applegate and Cook, 1991) and flow shop scheduling (Linn and Zhang, 1999). Typical application areas for solving this type of problems include warehouse location (Michel and Van Hentenryck, 2004), line balancing (Nilakantan et al., 2017), routing (Sitek and Wikarek, 2017);(Dang et al., 2012) or planning and scheduling technology design (Steger-Jensen et al., 2011).

The *No free lunch theorems* for optimisation have had a vital impact on design of efficient algorithms to solve combinatorial optimisation problems (Wolpert and Macready, 1997). One of the implications is that approximation capabilities and runtime of algorithms for combinatorial optimisation problems are now increasingly interpreted in their relation to specific problem instances.

In this paper, we present a computational study of local search strategies for the classical uncapacitated facility location problem (Aikens, 1985). This problem has very good scaling properties, as well as relatively less constrained search space that does not seem to change its structural properties heavily by scaling (Chalupa and Nielsen, 2018b).

Contributions. In this research we use two local search algorithms to solve the facility location problem and obtain that their performance comparison indicators depend vastly on the coefficients within the instance. We extend on a previous study of local search algorithms for very large instances of the problem (Chalupa and Nielsen, 2017) and consider this as part of the effort to develop robust local search algorithms (Chalupa and Nielsen, 2018a).

The first algorithm is steepest descent local search (LS), choosing the move to open or close a facility at each time step such that the best objective value is obtained. Another algorithm studied will be randomised local search (RLS), which attempts to open or close a facility and accepts the move whenever it does not lead to a worsening. We also use a mixed-integer linear programming solver to compare the results of the local search algorithms to distribution of the actual optima.

The experimental results are presented for four sets of problem instances, with each set containing instances with 1000 customers and numbers of facility sites ranging from 50 to 140 (for instances with up to 90 facility sites we also used the ILP-based solver). We obtain that the choice of the right local search strategy is indeed closely tied to

⋆ Full version of this article is available as a technical report: Chalupa, D., Nielsen, P. (2018). Instance Scale, Numerical Properties and Design of Metaheuristics: A Study for the Facility Location Problem. arXiv preprint, arXiv:1801.03419, *https://arxiv.org/abs/1801.03419*.

2405-8963
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2019.11.535

the numerical properties of a particular instance in strong support of the *No free lunch theorems*.

Further theoretical analyses may provide more rigorous explanations for some of our findings. Such analyses could pave the way to better understanding of the relation between instance structure and algorithm efficiency. However, these analyses are outside of scope of this particular article. On the other hand, the facility location problem belongs to a class of classical assignment and cost optimisation problems, i.e. similar phenomena may be discovered for other popular real-world combinatorial optimisation problems as well.

The paper is structured as follows. In Section 2, we introduce and review the uncapacitated facility location problem. In Section 3, we propose our four cost and distance models for instances of the problem, as well as the local search algorithms explored. Section 4 presents the experimental results and provides a brief discussion. Our conclusions are presented in Section 5.

2. THE UNCAPACITATED FACILITY LOCATION PROBLEM

We will firstly formalise the problem as an integer linear program and provide an overview of related results on heuristics and metaheuristics to solve the problem in large-scale.

Let $\mathcal{F}' \subseteq \mathcal{F}$ be a subset of selected potential facility locations. Let f_i be the cost of facility $i \in \mathcal{F}$ and let $d(j, i_j)$ be the distance from a customer $j \in \mathcal{C}$ to the nearest facility $i_j \in \mathcal{F}'$. Then, the objective in the uncapacitated facility location problem will be to minimise the following objective function:

$$\min \sum_{i \in \mathcal{F}'} f_i + \sum_{j \in \mathcal{C}} d(j, i_j). \tag{1}$$

It is possible to transform this formulation into an integer linear programming (ILP) formulation of the problem (Al-Sultan and Al-Fawzan, 1999). Let n be the number of facilities and let m be the number of customers. Then, alternatively, the objective is solve the following ILP formulation of the problem:

$$\min \sum_{i=1}^{n} f_i y_i + \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij}, \tag{2}$$

s.t.

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j = 1, ..., m \tag{3}$$

$$x_{ij} \leq y_i, \quad i = 1, ..., n, \quad j = 1, ..., m, \tag{4}$$

where c_{ij} is the cost of meeting the demand of customer j from facility i, and x_{ij} and y_i are binary decision variables determining if facility i is used to serve customer j and if a facility is established at position i.

With this exact formulation, an out-of-the-box mixed-integer linear programming solver can be used to solve the problem up to a certain scale. A brief study has previously been conducted for such an approach with a high number of customers (Chalupa and Nielsen, 2017).

We will investigate the performance of two local search algorithms for a set of carefully chosen facility location instances. These will highlight the contrast between search space structures and their influence on the actual behaviour of different optimisation techiniques. The aim is to establish how the behaviours of metaheuristics can vastly differ with only a slight variation in values within the problem instance. This will not only highlight the need for hybrid metaheuristics in these problems but will also strengthen the case for hybrid algorithms with a compact parameter suite and underline that robust approaches may for uncertain problem structures be preferable.

3. OUR FACILITY COST AND DISTANCE MODELS AND LOCAL SEARCH ALGORITHMS

In this section, we introduce our four facility cost and distance models, as well as the algorithms we use to solve them.

3.1 Facility Cost and Distance Models

Each of the four problem models have specific quantitative properties, leading to contrasting search landscapes. In Model 1, all facilities have the same cost, while the distances follow a moderately varied uniform distribution. Model 2 works with binary facility costs and a bimodal distribution of distances, with many distant connections but a few close ones. Model 3 assumes both binary facility costs and binary distances. Last but not least, Model 4 works with facility costs and distances following a Poissonian distribution.

Model 1: Flat Facility Cost, Moderately Varied Random Distances. The first model will assign the same unit cost to all facilities. The distances will be represented by integers taken uniformly at random from a limited interval between 1 and 10. This way plateaus will be generated in the search space. The corresponding values of f_i and c_{ij} will be:

$$f_i = 1, \quad i = 1, ...n, \tag{5}$$

$$c_{ij} = random(1, 10), \quad i = 1, ..., n, \quad j = 1, ..., m. \tag{6}$$

Model 2: Binary Facility Cost, Bimodal Distribution of Distances. In the second model, we use a binary choice facility cost. Each facility costs either 1 or 2 units. The distances will follow a bimodal distribution. Similarly to the previous model, we will generate uniformly random integer values between 1 and 10. However, each value higher than 1 will be overwritten by the maximum possible value. This generates instances with many high distances, but also several low distances between customers and facilities. The values within an instance will be the following:

$$f_i = random(1, 2), \quad i = 1, ...n, \tag{7}$$

$$c_{ij} = \begin{cases} 1 & with\ probability\ 0.1 \\ 10 & otherwise \end{cases}, \ i = 1, ..., n, \ j = 1, ..., m. \tag{8}$$

Model 3: Binary Facility Cost, Binary Distances. The facility cost structure in this model is the same as in the previous model. However, the distances will also be generated as 1 or 2. This will lead to a relative flat problem landscape. The aim will be to effectively search for a solution with as many facility cost and distance values equal to 1 as possible.

$$f_i = random(1,2), \quad i = 1,...n, \tag{9}$$

$$c_{ij} = random(1,2), \quad i = 1,...,n, \quad j = 1,...,m. \tag{10}$$

Model 4: Flat Facility Cost, Poissonian Distribution of Distances. The last model is a modification of Model 1. The uniform facility cost will be used, with each facility costing a single unit. The distances will be taken from the Poissonian distribution $Po_k(\lambda)$, with k trials and a probability of success λ/k per trial. The distance will be equal to the number of successful trials, incremented by 1, to avoid zero distance in case that all trials fail. This leads to a highly skewed distribution of distances. In our implementation, we choose $k = 10$ and $\lambda = 1$. The parameters of the model are the following:

$$f_i = 1, \quad i = 1,...n, \tag{11}$$

$$c_{ij} = 1 + Po_{k=10}(\lambda = 1), \quad i = 1,...,n, \quad j = 1,...,m. \tag{12}$$

3.2 Local Search Algorithms

We use two local search algorithms that have been explored in a previous study on a similar large-scale model of customer service centre applications (Chalupa and Nielsen, 2017). The algorithms have been chosen for their simplicity and scalability, as well as the fact that neither of them requires extensive explicit or implicit parameter tuning making them straight forward to implement.

The first one is a systematic local search (LS) algorithm that, at each time step, chooses the move that leads to the largest drop in the objective value. The second algorithm is a randomised local search (RLS), which attempts opening or closing a single randomly chosen facility in each time step and accepts the move if it does not lead to a worsening.

Local search (LS). The algorithm will search in space of bit strings, i.e. in $\{0,1\}^n$. Each bit y_i determines whether facility i is open or closed, similarly to the ILP formulation of the problem. The algorithm starts with all facilities open, i.e. with a bit string consisting solely of 1-bits. Each customer is then assigned to the closest facility. Let y be the current bit string and let $y'(i)$ be the bit string obtained by flipping bit y_i in y. Then, LS chooses i such that the objective value of $y'(i)$ is minimised within the neighbourhood. In our investigations, LS will be terminated after n iterations, where n is the number of candidate facilities.

Randomised local search (RLS). This algorithm also starts with all facilities open. Let y be the current bit string and let $y'(i)$ be the bit string obtained by flipping bit y_i in y. Then, RLS chooses $1 \leq i \leq n$ at random in each time step. The new bit string $y'(i)$ is accepted if the objective value of $y'(i)$ is not higher than the objective value of y. In terms of the number of bit flips attempted, each iteration of LS corresponds to n iterations of RLS. RLS will therefore be terminated after n^2 iterations in our further experiments.

4. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we present the experimental results obtained. We first describe the experimental settings and then present the results for the four problem instance models.

4.1 Experimental Settings

For each of the facility cost and distance models, we generated 10 instances and ran LS and RLS 1000 times for each instance.

To determine actual optima or near-optimal solution we use the CBC mixed-integer programming branch-and-cut solver from the COIN-OR package (Bonami et al., 2008; Linderoth and Lodi, 2011). We use a precompiled 64-bit Windows binary of CBC compiled by the Intel 11.1 compiler. The time limit for CBC solution search was set to 24 hours per instance.

The experiments were performed on a machine with an Intel Core i7-6820 CPU @ 2.70 GHz, 32 GB RAM and Windows 10 operating system. LS and RLS were implemented in C++ using Qt, compiled by the MinGW 32-bit compiler.

Each run of LS was stopped after n iterations, as in each iteration, a flip of each bit separately is attempted. For RLS, n^2 local search iterations were allowed, since a single bit is always flipped in each iteration.

4.2 Results for the Four Facility Cost and Distance Models

The experimental results are presented for each model separately using box-whisker plots to illustrate the distribution of solutions found by LS and RLS. The distribution of optimal or near-optimal solutions and lower bounds will also be presented.

Model 1: Flat Facility Cost, Moderately Varied Random Distances. Figure 1 presents the box-whisker plots obtained for the instances generated by Model 1. For these instances, RLS outperforms LS. This can be explained by the possibility that closing or opening a facility, which leads to a higher objective value at the moment, could not necessarily lead to the best moves in the future. CBC was able to find optima for instances with up to 60 customers, as well as for most instances with 70 customers. One can observe that the gaps between the typical performances tend to widen with growing instance size. This suggests that more advanced techniques could offer some improvement in this model, including tabu search (Sun, 2006) or evolutionary algorithms (Jaramillo et al., 2002).

While the numerical differences observed here are not that large, it is worth mentioning that more pronounced differences between LS and RLS are obtainable. This is

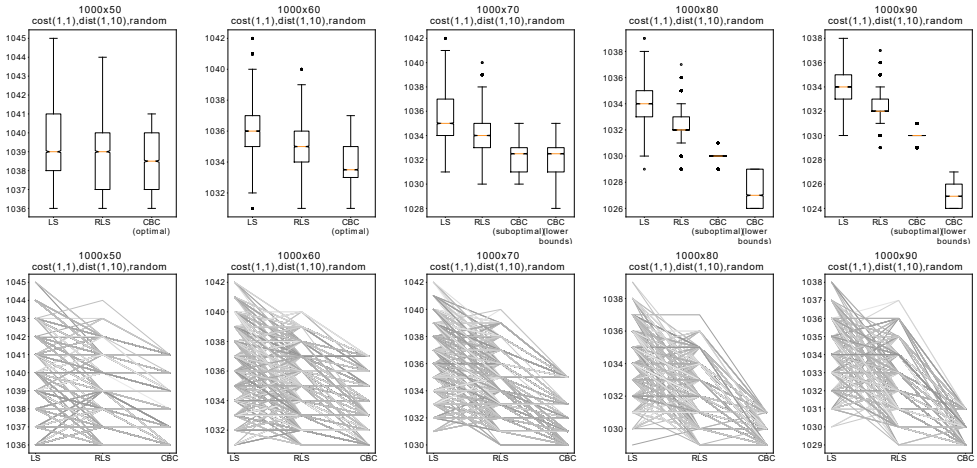

Fig. 1. Box-whisker plots and plots depicting the relation of the performance of LS, RLS and the actual optimal or near-optimal solutions obtained for 10 problem instances generated according to Model 1. LS and RLS were used 1000 times per instance, while the results for CBC represent optimal or near-optimal reference solutions obtained by the corresponding ILP solving procedure.

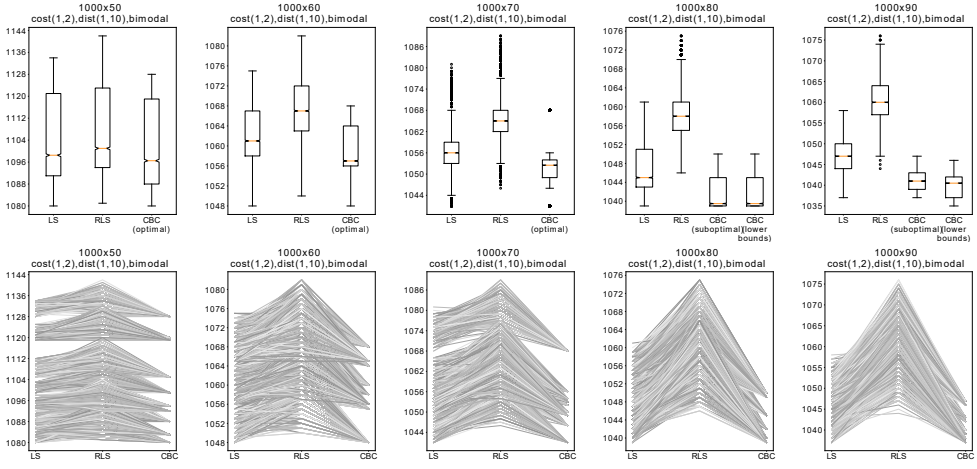

Fig. 2. Box-whisker plots and plots depicting the relation of the performance of LS, RLS and the actual optimal or near-optimal solutions obtained for 10 problem instances generated according to Model 2. LS and RLS were used 1000 times per instance, while the results for CBC represent optimal or near-optimal reference solutions obtained by the corresponding ILP solving procedure.

possible by changing the value we have fixed to 10 for simplicity of this study. We have picked this approach for simplicity and to obtain results that are easy to interpret in a broader context.

Figure 1 also sheds more light on the relations between the performances of different techniques, grouped by instances. Each line in the plot connects results for the corresponding runs of LS, RLS and CBC, linking the solutions found by the local search algorithms to the actual optimal or near-optimal solutions. The aim of these plots is to explore the variability of the results obtained on a per-instance level, in addition to the overall aggregation explored in the box-whisker plots.

One can observe an overall "downward" trend from LS to RLS, confirming the better performance of RLS also on the per-instance level. This needs to be interpreted in context: the fact that the performance of an individual run of LS or RLS is inferior does not necessarily mean that a multi-start variant of the algorithm cannot succeed.

Model 2: Binary Facility Cost, Bimodal Distribution of Distances. Figure 2 illustrates the results obtained for Model 2 as box-wisker plots. These reveal a contrasting pattern to the one obtained for Model 1. LS outperforms RLS for these instances. At this point, it is worth noting that we have only changed the cost and distance structure. Such a change already leads to a very different result to the one observed for the previous instances. Observing the distributions of results found by LS and RLS, one can see that the gap even widens with growing number of facilities. However, Figure 2 does not seem to indicate a pronounced slope in the relations between objective values observed for high-quality runs of LS and the results found by CBC.

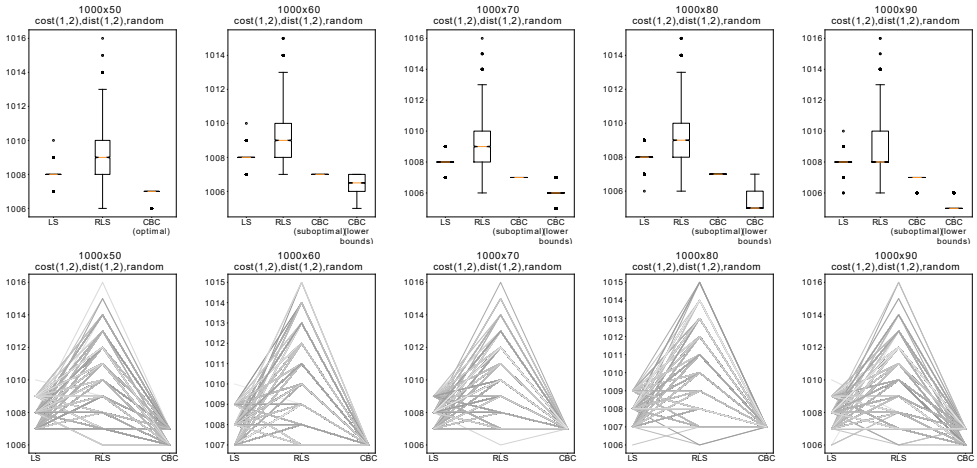

Fig. 3. Box-whisker plots and plots depicting the relation of the performance of LS, RLS and the actual optimal or near-optimal solutions obtained for 10 problem instances generated according to Model 3. LS and RLS were used 1000 times per instance, while the results for CBC represent optimal or near-optimal reference solutions obtained by the corresponding ILP solving procedure.

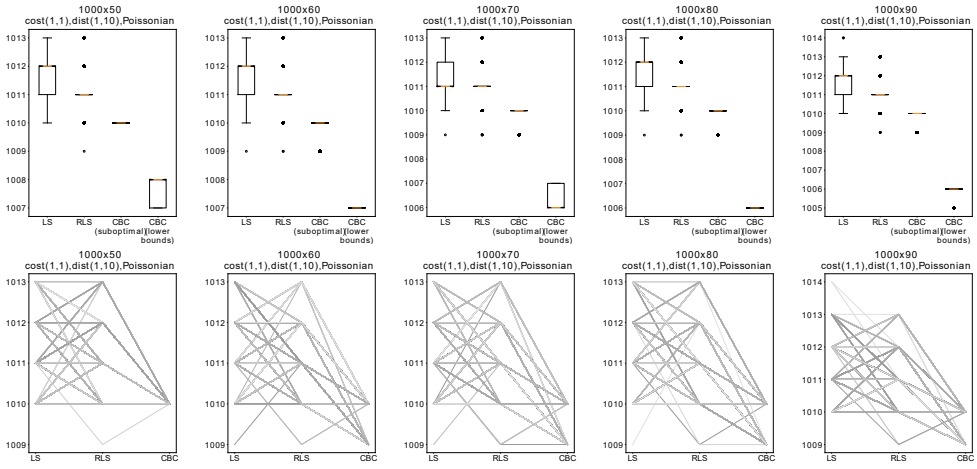

Fig. 4. Box-whisker plots and plots depicting the relation of the performance of LS, RLS and the actual optimal or near-optimal solutions obtained for 10 problem instances generated according to Model 4. LS and RLS were used 1000 times per instance, while the results for CBC represent optimal or near-optimal reference solutions obtained by the corresponding ILP solving procedure.

Model 3: Binary Facility Cost, Binary Distances. Figure 3 reveals the results obtained for Model 3. These paint a more complex picture than those obtained for the previous instances. This is further supported by the results of CBC, which was able to find proven optima only for the instances with 50 facilities. For these instances, LS has a very intriguing performance. One can observe that the median solution quality is better than for RLS. The distribution of solution quality for LS seems to be estimated relatively well in terms of its shape. However, one can also observe a consistent bias in this estimate. The results obtained by LS are heavily concentrated around the median, similarly to the distribution of the actual optimal or near-optimal solutions.

A surprising observation is that while RLS often has inferior performance, it can actually outperform LS at times due to its randomised nature. Model 3 therefore represents an instance type, for which LS is a good choice to rapidly obtain a good but not necessarily optimal solution. If one needs to obtain a near-optimal solution, then RLS might be a better choice of an algorithm due to its randomised nature and better robustness.

Model 4: Flat Facility Cost, Poissonian Distribution of Distances. Figure 4 presents the results for the last model in the form of box-whisker plots. This model seems to be the most intriguing one out of the four models investigated. In contrast to Model 3, it is RLS, for which the results are concentrated around the median. However, RLS still seems to perform better than LS in most cases, even though the results indicate that it also has a tendency to get stuck in local optima.

The patterns presented in Figure 4 also reveal that in most cases, LS and RLS produce solutions of comparable quality. RLS is therefore likely to sample solutions with better objective values more frequently. However, for the

1000×50 instances, one can also observe that while CBC consistently produced solutions with objective value 1010, RLS was able to provide a better solution with objective value 1009. These results have indeed painted a very complex picture of problem difficulty landscape and algorithm performance.

5. CONCLUSIONS

We proposed four cost and distance models for the uncapacitated facility location problem instances. These models enabled us to uncover a complex relation between the numerical properties of a problem instance and efficiency of the optimisation algorithms used to solve the problem. An investigation of the efficiency of two metaheuristic algorithms was presented, namely systematic local search (LS) and randomised local search (RLS). An out-of-the-box mixed-integer linear programming solver has also been used to obtain reference results. In the experimental results, one can observe a rich variety of behaviours. RLS outperformed LS for Model 1, while this was reversed in Model 2. Models 3 and 4 show even more intricate landscape properties and behaviours of LS, RLS, as well as the exact solver.

Based on the results obtained by LS and RLS, one can conclude that the choice of the right local search strategy for a particular model is closely related to its internal numerical properties. This strengthens the case for design of hybrid metaheuristics for this type of problems, but also highlights the intriguing need for the numerical properties of a particular instance to be taken into account in such a design.

Further theoretical insights may also be of a high interest, particularly into the reasons behind the relation between the instance scale, structure, and the behaviour of optimisation algorithms. Such an understand may open the way to new results and generalisations to other related assignment problems and combinatorial problems in general, such as job shop and flow shop scheduling, knapsack problems or pallet stacking.

REFERENCES

Aikens, C.H. (1985). Facility location models for distribution planning. *European Journal of Operational Research*, 22(3), 263–279.

Al-Sultan, K.S. and Al-Fawzan, M.A. (1999). A tabu search approach to the uncapacitated facility location problem. *Annals of Operations Research*, 86, 91–103.

Applegate, D. and Cook, W. (1991). A computational study of the job-shop scheduling problem. *ORSA Journal on Computing*, 3(2), 149–156.

Blum, C. and Sampels, M. (2004). An ant colony optimization algorithm for shop scheduling problems. *Journal of Mathematical Modelling and Algorithms*, 3(3), 285–308.

Bonami, P., Biegler, L.T., Conn, A.R., Cornuéjols, G., Grossmann, I.E., Laird, C. D.and Lee, J., Lodi, A., Margot, F., Sawaya, N., et al. (2008). An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2), 186–204.

Chalupa, D. and Nielsen, P. (2017). A large-scale customer-facility network model for customer service centre location applications. Technical report, Operations Research, Aalborg University, Fibigerstræde 16, 9220 Aalborg.

Chalupa, D. and Nielsen, P. (2018a). A simple and robust monte carlo hybrid local search algorithm for the facility location problem. *Engineering Optimization*. doi:10.1080/0305215X.2018.1497618. Article in Press.

Chalupa, D. and Nielsen, P. (2018b). Two strategies of two-level facility network design for autonomous ground vehicle operations. *Production and Manufacturing Research*, 6(1), 494–506. doi: 10.1080/21693277.2018.1548982.

Dang, Q.V., Nielsen, I., and Bocewicz, G. (2012). A genetic algorithm-based heuristic for part-feeding mobile robot scheduling problem. *Advances in Intelligent and Soft Computing*, 157 AISC, 85–92. doi:10.1007/978-3-642-28795-4$_1$0.

Gao, L.L. and Robinson, E.P. (1992). A dual-based optimization procedure for the two-echelon uncapacitated facility location problem. *Naval Research Logistics (NRL)*, 39(2), 191–212.

Jaramillo, J.H., Bhadury, J., and Batta, R. (2002). On the use of genetic algorithms to solve location problems. *Computers & Operations Research*, 29(6), 761–779.

Linderoth, J.T. and Lodi, A. (2011). MILP software. *Wiley encyclopedia of operations research and management science*.

Linn, R. and Zhang, W. (1999). Hybrid flow shop scheduling: a survey. *Computers & industrial engineering*, 37(1-2), 57–61.

Melo, M.T., Nickel, S., and Saldanha-Da-Gama, F. (2009). Facility location and supply chain management–a review. *European Journal of Operational Research*, 196(2), 401–412.

Michel, L. and Van Hentenryck, P. (2004). A simple tabu search for warehouse location. *European Journal of Operational Research*, 157(3), 576–591.

Nilakantan, J., Li, Z., Tang, Q., and Nielsen, P. (2017). Multi-objective co-operative co-evolutionary algorithm for minimizing carbon footprint and maximizing line efficiency in robotic assembly line systems. *Journal of Cleaner Production*, 156, 124–136. doi:10.1016/j.jclepro.2017.04.032.

Sitek, P., Nielsen, I., and Wikarek, J. (2014). A hybrid multi-agent approach to the solving supply chain problems. volume 35, 1557–1566. doi:10.1016/j.procs.2014.08.239.

Sitek, P. and Wikarek, J. (2017). Capacitated vehicle routing problem with pick-up and alternative delivery (cvrppad): model and implementation using hybrid approach. *Annals of Operations Research*, 1–21. doi:10.1007/s10479-017-2722-x. Article in Press.

Steger-Jensen, K., Hvolby, H.H., Nielsen, P., and Nielsen, I. (2011). Advanced planning and scheduling technology. *Production Planning & Control*, 22(8), 800–808.

Sun, M. (2006). Solving the uncapacitated facility location problem using tabu search. *Computers & Operations Research*, 33(9), 2563–2589.

Wolpert, D.H. and Macready, W.G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67–82.